**Start** -> func int var initializer: begin A return var end Start | ɛ

**A** -> begin A end | if EXP1: begin A end **B** A |for Initializer , EXP1 , Initializer2: begin A end A |

print ID ; A | print ‘str’; A | in variable ; A | Initializer A | return ID |

**B -> el**if EXP1: begin A end B | C

**C->**  else: begin A end

**Initializer** -> var <- EXP2 int ; **Initializer** | var int ; **Initializer** | Initializer2, **Initializer** | var ;

| var , **Initializer** | ;

**Initializer2** -> ID <- EXP2

**EXP1** -> E RO E

**E** -> T E'

**E'** -> + T E' | - T E' | ^

**T** -> F T'

**T'** -> \* F T' | / F T' | ^

**F** -> id | num | E

**RO** -> == | != | > | >= | < | <=

**ID** -> var | num

**Str** -> (characters | integers) A | ɛ

E -> E + T | E - T | T

T -> T \* F | T / F | F

F -> id | num | ( E )

**EXP2** -> EXP3 + EXP2 | EXP3 – EXP2 | EXP3

**EXP3** -> ID \* EXP3 | ID / EXP3 | ID